International Journal of Advances in Engineering and Management (IJAEM)
Volume 2, Issue 9, pp: 466-472 www.ijaem.net ISSN: 2395-5252

# A Comprehensive Analysis of Disk Scheduling Algorithms

Ahalya R, Chethana S, Varsha A, Dr.M. Sujithra M.C.A, M.Phil., PhD, Dr.A.D. Chitra M.C.A, M.Phil., PhD,

*2nd Year, M.Sc. Software Systems (Integrated), Coimbatore Institute of Technology, Coimbatore*
*Assistant Professor, Department of Data Science, Coimbatore Institute of Technology, Coimbatore*
*Assistant Professor, Department of Software Systems, Coimbatore Institute of Technology, Coimbatore*

Date of Submission: 15-11-2020 Date of Acceptance: 28-11-2020

**ABSTRACT:** The rise in the speed of processors and main memory over the years, has surpassed that of disk access, with processor and main memory speeds increasing about twice to that of the speed of the disk. The result is that the disks are currently at least four times slower than main memory. Hence, the performance of disk storage subsystem is of vital importance, and much researches are underway for improving this performance. In OS, seek time is the important parameter to get the best access time. So, disk schedulingis one of the most important responsibilities of the operating systems.

## I. INTRODUCTION

A process needs two types of time- CPU time and I/O time. I/Orequests the OS to access the disk. The OS must be fair enough to satisfy each request. The technique that OS employs to overcome this is disk scheduling- to determine the request which is to be satisfied next. In any disk system with a dynamic read/write head, the seek time between cylinders takes a notable amount of time. This seek timeshould be minimized to get better access time. The main goals of disk scheduling are:

- Fairness
- High throughout
- Minimal traveling head time

**Parameters Defining Disk Scheduling**

**1. Seek Time** -- The time taken for the disk arm to move to the desired track or cylinder.

$$\boxed{T(S)=m*n+s}$$

where
n = number of tracks,
m = constant track crossing rate [m can range from 0.1 to 0.3 ms]
s =acceleration and deceleration time [s can range from 3 to 20 ms]

**2. Rotational Delay (Latency)** – Thetime taken for the addressed sector of the disk to spin into a position where it is accessible by the read/write head.

$$\boxed{T(latency)=1/2*1/r}$$

**3. Transfer Time** -- The transfer time to or from the depends on the rotation speed of the disk.

$$\boxed{T(transfer)=b/(r*N)}$$

where
T = transfer time,
b = number of bytes to be transferred,
N = number of bytes on a track, and
r = rotation speed, in revolutions per second.

**4. Access Time--** It is the sum of the above three factors

$$\boxed{T(access) =T (seek)+T(latency)+T(transfer)}$$

**WHY DO WE NEED TO SCHEDULE A DISK?**

The main purpose of disk scheduling algorithm is to select a disk request from the queue of I/O requests and to decide this request will be processed. Scheduling is very essential in reducing the total seek time.

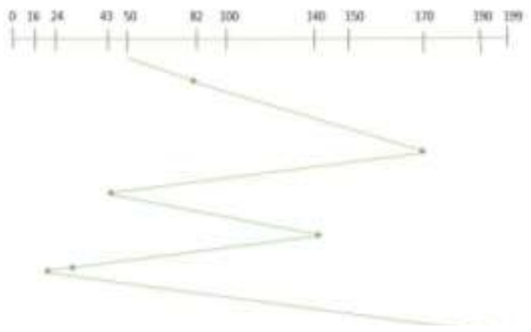
**DISK SCHEDULING ALGORITHMS**

The various disks scheduling algorithm are discussed below. Each algorithm has some advantages and disadvantages. The limitation of each algorithm leads to the evolution of a new algorithm.

**FCFS or First-come-first-serve Scheduling:**

First-come-first-serve Schedulingis the simplest of all the disk scheduling policies. In this, the requests are addressed in the order they arrive in the disk queue.

Take the following sequence--82, 170,43, 140, 24,16,190 with the Read-write head initially at the track 50 and the tail track being at 199.



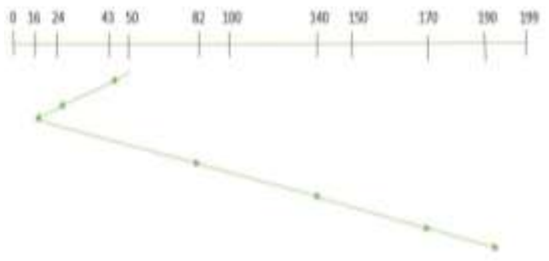
Total seektime is calculated as,

=(82-50)+(170-82)+(170-43)+(140-43)+(140-24)+(24-16)+(190-16)
= 642 ms

As we can see, every request gets a fair chance in FCFS policy and also there is no indefinite delay. But this algorithm does not try to optimize the total seek time and also it may not provide the best possible service.

**SSTF or Shortest Seek Time First Scheduling:**

SSTF algorithm selects theI/O request which requires the minimum arm movement from its current position. It greatly reduces the total seek time when compared to FCFS. It allows the disk head to move to the closest track in the service queue. Taking the same example as before,



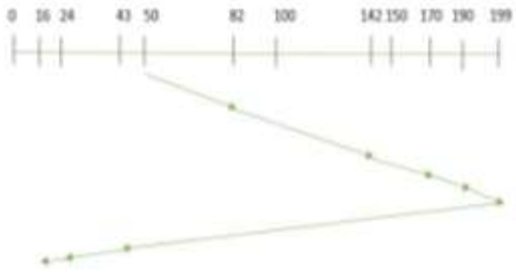
So, total seek time:
=(50-43)+(43-24)+(24-16)+(82-16)+(140-82)+(170-40)+(190-170)
= 208 ms

SSTF significantly decreases the average Response Time and hence Throughput increases. While we will have to suffer the overhead to calculate seek time in advance and it can also cause Starvation for a request if it has higher seek time as compared to incoming requests

**SCAN or Elevator Scheduling:**

In SCAN algorithm, the disk arm moves in a specific direction till the end, serving all the requests coming in its path,and then it turns backand moves in the reverse direction serving the requests coming in its path.It works in the same way as an elevator, that is, an elevator moves completely till the last floor of a particular direction and then turns back. Considering the same example as before,

The requests to be addressed are--82,170,43,140,24,16,190. The Read/Write arm is at 50.



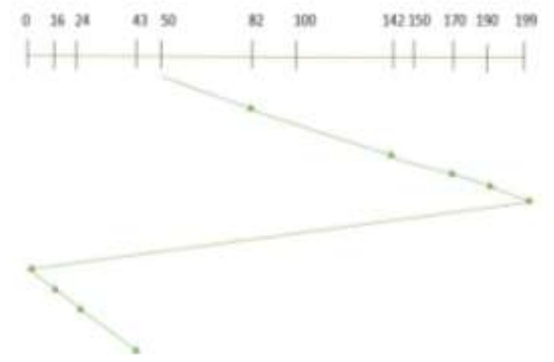
The total seek time is calculated as,
=(199-50)+(199-16)
= 332 ms

As we can see, the requests at the midrange are serviced more and those arriving behind the disk arm will have to wait. SCAN algorithm ensures high throughput and Low variance of response time

**C-SCAN or Circular SCAN Scheduling:**

In C-SCAN, the disk arm moves in a specific direction satisfying all the requests in its path, until it reaches the end, and then it jumps to the otherend of the opposite direction without satisfyingany requests and then again, it turns back and startsto move in that direction satisfying the remaining requests. So, the disk arm moves in a circular motion and thisissimilar to SCAN algorithm and hence it is known as C-SCAN.

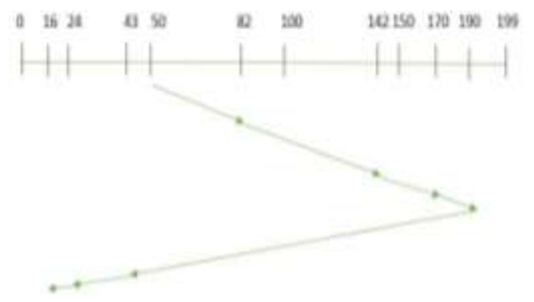
Consider the requests to be addressed are--82,170,43,140,24,16,190. Initially the Read/Write arm is at 50.



Seek time is calculated as:
=(199-50)+(199-0)+(43-0)
=391 ms
C-SCAN algorithm provides more uniform wait time compared to SCAN

**LOOK Scheduling:**

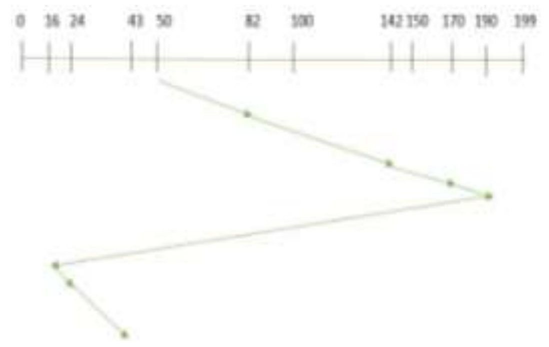
LOOK is similar to that of SCANalgorithm except that the disk arm instead of going till the end of the disk, goes only to the last request to be satisfied in front of the head and only then reverses its direction from there. Hence, it prevents the extra delay which occurs due to unnecessary traversal till the end of the disk. Considering the same example,



So, the seek time is calculated as:
=(190-50)+(190-16)
=314 ms

**C-LOOK or Circular LOOK Scheduling:**

In a way, C-LOOK is similar to C-SCAN algorithm. In C-LOOK, the disk armgoes only till the last request to be satisfied in front of the head, then it jumps to the lowest request cylinder without satisfying any requests and then it again starts moving outwards satisfying the remaining requests.It is different from C-SCAN because, it forces the disk arm to move till the last cylinder regardless of knowing whether any request is to be serviced on that cylinder or not. Taking the same example,

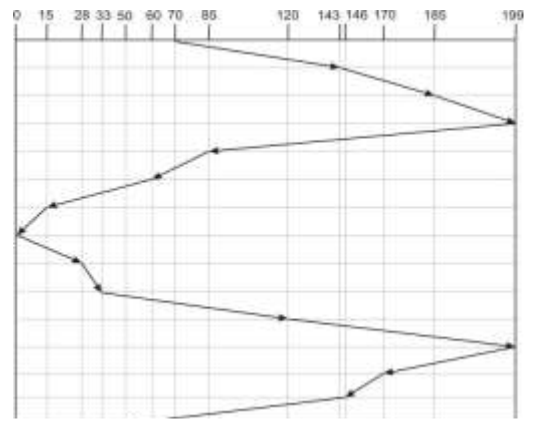


So, the seek time is calculated as:
=(190-50)+(190-16)+(43-16)
=341 ms

**N-STEP SCAN Scheduling:**

N-STEP SCANor otherwise known asN-STEP LOOK algorithm segments the disk requests into sub queues which are of length N. After that, all the sub queues are processed one at a time using SCAN policy. If suppose there requests less than N at the end of a scan, then all of those requests are processes in a single go with the next scan. This policy eliminates starvation of requestscompletely. N-Step SCAN can be adjusted by varying the value ofN. When N=1, N-Step SCAN degenerates to FCFS. IfNbecomes infinite, then N-Step degenerates to SCAN.

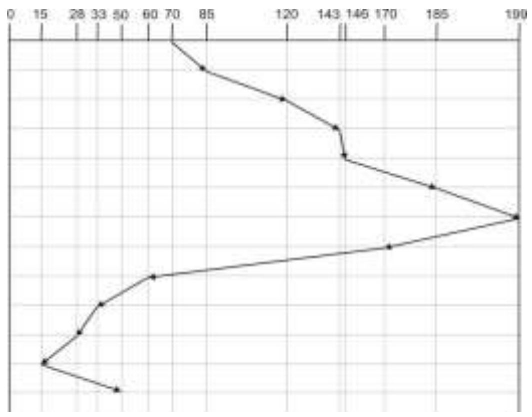
Suppose therequests to be addressed are--60, 143, 15, 185, 85, 120, 33, 28, 146. Assuming that the read/write head is initially at cylinder 70.



**FSCAN or Freezing-SCAN Scheduling:**

FSCAN is modification of SCAN algorithm.This policy uses two sub queues.When the scan begins, all of the existing requests are put into the first queue, while all new requests are put into the second queue. Service of new requests is delayed until all of the old requests have been satisfied. Hence, its name Freezing SCAN (FSCAN).

Suppose therequests to be addressed are--60, 143, 15, 185, 85, 120, 33, 28, 146. The read/write head is initially at cylinder 70.



**Selecting A Disk Scheduling Algorithm**

SSTF algorithm is very common and has its appeal because it has an increased performance over FCFS. SCAN and C-SCAN perform better for systems that workson a heavy load and rarely cause starvation. Commonly, SSTF or LOOK is a reasonable choice for the default algorithm.
The performance of these algorithms depends on the number and types of requests.

**COMPARISON OF THE VARIOUS DISK SCHEDULING ALGORITHMS**

compares the average head movement of six disk scheduling algorithms for the first five runs and their average. Similar requests are assigned for every individual run for all six algorithms and their total head movement is calculated.

| S.No. | FCFS | SSTF | Scan | Look | C-Scan | C-Look |
|---|---|---|---|---|---|---|
| 1 | 540 | 240 | 290 | 240 | 363 | 295 |
| 2 | 631 | 276 | 280 | 276 | 348 | 306 |
| 3 | 264 | 217 | 270 | 224 | 393 | 289 |
| 4 | 322 | 189 | 235 | 189 | 363 | 189 |
| 5 | 640 | 235 | 275 | 245 | 378 | 300 |
| Average | 479 | 231 | 270 | 235 | 369 | 276 |

**Table 1:** Average head movement of first five runs and their average.

Shortest Seek Time First algorithm produces the minimum head movement of 240 in the first run, 276 in the second run, 217 in the third run, 189 in the fourth run, 235 in the last run and average head movement of all the runs is 231.From the above table, it is very much clear that for the different five runs, the Shortest Seek Time First disk scheduling algorithm has minimum total head movement in all cases as compared to other five algorithms. A graphical representation is shown in the figure 8 with the help of data evaluated in table 1. ompares the average head movement of six disk scheduling algorithms for the first five runs and their average. Similar requests are assigned for every individual run for all six algorithms and their total head movement is calculated.

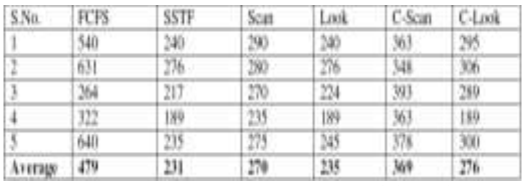
S.No.
FCFS
SSTF
Scan
Look
C-Scan

C-Look
1
540
240
290
240
363
295
2
631
276
280
276
348
306
3
264
217
270
224
393
289
4
322
189
235
189
363
189
5
640
235
275
245
378
300
Average
479
231
270
235
369
276

**Table 1:** Average head movement of first five runs and their average.

Shortest Seek Time First algorithm produces the minimum head movement of 240 in the first run, 276 in the second run, 217 in the third run, 189 in the fourth run, 235 in the last run and average head movement of all the runs is 231. From the above table, it is very much clear that for the different five runs, the Shortest Seek Time First disk scheduling algorithm has minimum total head movement in all cases as compared to other five algorithms. A graphical representation is shown in the figure 8 with the help of data evaluated in table 1

Table 1 given below compares the average head movement of six disk scheduling algorithms for the first five runs and their average. Similar requests (as seen previously) are assigned for every individual run for all six algorithms and their total head movement is calculated.



| S.No. | FCFS | SSTF | Scan | Look | C-Scan | C-Look |
|---|---|---|---|---|---|---|
| 1 | 540 | 240 | 290 | 240 | 363 | 295 |
| 2 | 631 | 276 | 280 | 276 | 348 | 306 |
| 3 | 264 | 217 | 270 | 224 | 393 | 289 |
| 4 | 322 | 189 | 235 | 189 | 363 | 189 |
| 5 | 640 | 235 | 275 | 245 | 378 | 300 |
| Average | 479 | 231 | 270 | 235 | 369 | 276 |

**Table 1:** Average head movement of first five runs and their average.

Shortest Seek Time First algorithm produces the minimum head movement of 240 in the first run, 276 in the second run, 217 in the third run, 189 in the fourth run, 235 in the last run and average head movement of all the runs is 231. From the above table, it is very much clear that for the different five runs, the Shortest Seek Time First disk scheduling algorithm has minimum total head movement in all cases as compared to other five algorithms. Thus, SSTF algorithm is a good criterion for selecting the requests for I/O from the disk queue.

## II. CONCLUSION

The performance of disk scheduling algorithm depends solely on the number of head movement, seek time and rotational latency. Requests for disk service are influenced by the file-allocation methods. The disk-scheduling algorithm should be written as a separate section of the OS, thus allowing it to be replaced with a different algorithm (if necessary). Few algorithms like SSTF and LOOK will be the most efficient algorithm compared to FCFS, SCAN, C-SCAN and C-LOOK disk scheduling algorithms with respect to the performance parameters.compares the average head movement of six disk scheduling algorithms for the first five runs and their average. Similar requests are assigned for every individual run for all six algorithms and their total head movement is calculated.
S.No.
FCFS
SSTF
Scan

| | | | | Look | C-Scan | C-Look |
|---|---|---|---|---|---|---|
| 1 | 540 | 240 | 290 | 240 | 363 | 295 |
| 2 | 631 | 276 | 280 | 276 | 348 | 306 |
| 3 | 264 | 217 | 270 | 224 | 393 | 289 |
| 4 | 322 | 189 | 235 | 189 | 363 | 189 |
| 5 | 640 | 235 | 275 | 245 | 378 | 300 |
| Average | 479 | 231 | 270 | 235 | 369 | 276 |

**Table 1:** Average head movement of first five runs and their average.

Shortest Seek Time First algorithm produces the minimum head movement of 240 in the first run, 276 in the second run, 217 in the third run, 189 in the fourth run, 235 in the last run and average head movement of all the runs is 231. From the above table, it is very much clear that for the different five runs, the Shortest Seek Time First disk scheduling algorithm has minimum total head movement in all cases as compared to other five algorithms. A graphical representation is shown in the figure 8 with the help of data evaluated in table 1.compares the average head movement of six disk scheduling algorithms for the first five runs and their average. Similar requests are assigned for every individual run for all six algorithms and their total head movement is calculated.

| S.No. | FCFS | SSTF | Scan | Look | C-Scan | C-Look |
|---|---|---|---|---|---|---|
| 1 | 540 | 240 | 290 | 240 | 363 | 295 |
| 2 | 631 | 276 | 280 | 276 | 348 | 306 |
| 3 | 264 | 217 | 270 | 224 | 393 | 289 |
| 4 | 322 | 189 | 235 | 189 | 363 | 189 |
| 5 | 640 | 235 | 275 | 245 | 378 | 300 |
| Average | 479 | 231 | 270 | 235 | 369 | |

276

Table 1: Average head movement of first five runs and their average.

Shortest Seek Time First algorithm produces the minimum head movement of 240 in the first run, 276 in the second run, 217 in the third run, 189 in the fourth run, 235 in the last run and average head movement of all the runs is 231. From the above table, it is very much clear that for the different five runs, the Shortest Seek Time First disk scheduling algorithm has minimum total head movement in all cases as compared to other five algorithms. A graphical representation is shown in the figure 8 with the help of data evaluated in table 1.compares the average head movement of six disk scheduling algorithms for the first five runs and their average. Similar requests are assigned for every individual run for all six algorithms and their total head movement is calculated.

| S.No. | FCFS | SSTF | Scan | Look | C-Scan | C-Look |
|---|---|---|---|---|---|---|
| 1 | 540 | 240 | 290 | 240 | 363 | 295 |
| 2 | 631 | 276 | 280 | 276 | 348 | 306 |
| 3 | 264 | 217 | 270 | 224 | 393 | 289 |
| 4 | 322 | 189 | 235 | 189 | 363 | 189 |
| 5 | 640 | 235 | 275 | 245 | 378 | 300 |
| Average | 479 | 231 | 270 | 235 | 369 | 276 |

Table 1: Average head movement of first five runs and their average.

Shortest Seek Time First algorithm produces the minimum head movement of 240 in the first run, 276 in the second run, 217 in the third run, 189 in the fourth run, 235 in the last run and average head movement of all the runs is 231. From the above table, it is very much clear that for the different five runs, the Shortest Seek Time First disk scheduling algorithm has minimum total head movement in all cases as compared to other five algorithms. A graphical representation is shown in the figure 8 with the help of data evaluated in table 1.

## REFERENCE

[1]. William Stallings, Operating Systems, Internal and Design Principles, 9th Edition, Dorling Kindersley Pvt. Ltd., 2018
[2]. AComparative Study of Disk Scheduling Algorithms-International Journal of Computer Science Trends and Technology (IJCST) – Volume 4 Issue 1, Jan - Feb 2016
[3]. Silberschatz A., Peterson J.L and Galvin P., "Operating System Concepts", John Wiley Publishing Company, 2002.
[4]. H.M. Deital, " An introduction to Operating System", Pearson Education, 2001
[5]. https://www.geeksforgeeks.org/disk-scheduling-algorithms/
[6]. https://www.javatpoint.com/os-disk-scheduling
[7]. Charles Crowley, "Operating System a Design Oriented Approach", Tata McGraw Hill, 2000.